

# Лекция 1

ПМ.02 Разработка, адаптация и  
внедрение ПО отраслевой  
направленности

МДК 02.01 Раздел 2

Основы программирования  
информационного контента на  
ЯВУ

Тимашева Эльза Ринадовна

Вид учебной работы	Объем часов	
	5 семестр	6 семестр
Максимальная учебная нагрузка (всего)		
Обязательная аудиторная учебная нагрузка (всего)		
в том числе:		
лекции	18 (9ПАР)	32 (16ПАР)
лабораторные занятия	14 (7ПАР)	28 (14ПАР)
Самостоятельная работа обучающегося (всего)		
в том числе:		
Консультации		
Форма промежуточной аттестации	Зачет	зачет

# Язык C# и платформа .NET

Первая версия языка вышла вместе с релизом Microsoft Visual Studio [.NET](#) в феврале 2002 года. Текущей версией языка является версия C# 10.0, которая вышла в ноябре 2021 года вместе с релизом .NET 6.

C# является языком с Си-подобным синтаксисом и близок в этом отношении к C++ и Java.

C# является объектно-ориентированным и в этом плане много

# Роль платформы .NET

Когда говорят C#, нередко имеют в виду технологии платформы .NET (Windows Forms, WPF, ASP.NET, Xamarin). И, наоборот, когда говорят .NET, нередко имеют в виду C#. Однако, хотя эти понятия связаны, отождествлять их неверно. Язык C# был создан специально для работы с фреймворком .NET, однако само понятие .NET несколько шире.

Фреймворк .NET представляет мощную платформу для создания приложений. Можно выделить следующие ее основные черты:

- **Поддержка нескольких языков.** Основой платформы является общезыковая среда исполнения Common Language Runtime (CLR), благодаря чему .NET поддерживает несколько языков: наряду с C# это также VB.NET, C++, F#, а также различные диалекты других языков, привязанные к .NET, например, Delphi.NET. При компиляции код на любом из этих языков компилируется в сборку на общем языке CIL (Common Intermediate Language) - своего рода ассемблер платформы .NET. Поэтому мы можем сделать отдельные модули одного приложения на отдельных языках.
- **Кроссплатформенность.** .NET является переносимой платформой (с некоторыми ограничениями). Например, последняя версия платформы на данный момент .NET Core поддерживается на большинстве современных ОС Windows, MacOS, Linux. Используя различные технологии на платформе .NET, можно разрабатывать приложения на языке C# для самых разных платформ - Windows, MacOS, Linux, Android, iOS, Tizen.

# Роль платформы .NET

- **Мощная библиотека классов.** .NET представляет единую для всех поддерживаемых языков библиотеку классов. И какое бы приложение мы не собирались писать на С# - текстовый редактор, чат или сложный веб-сайт - так или иначе мы задействуем библиотеку классов .NET.
- **Разнообразие технологий.** Общеязыковая среда исполнения CLR и базовая библиотека классов являются основой для целого стека технологий, которые разработчики могут задействовать при построении тех или иных приложений. Например, для работы с базами данных в этом стеке технологий предназначена технология ADO.NET и Entity Framework Core. Для построения графических приложений с богатым насыщенным интерфейсом - технология WPF и UWP, для создания более простых графических приложений - Windows Forms. Для разработки мобильных приложений - Xamarin. Для создания веб-сайтов - ASP.NET и т.д.

Также еще следует отметить такую особенность языка С# и фреймворка .NET, как автоматическая сборка мусора. А это значит, что нам в большинстве случаев не придется, в отличие от С++, заботиться об освобождении памяти. Вышеупомянутая общеязыковая среда CLR сама вызовет сборщик мусора и очистит память.

# .NET Framework и .NET Core

.NET долгое время развивался преимущественно как платформа для Windows под названием .NET Framework. В 2019 вышла последняя версия этой платформы - .NET Framework 4.8. Она больше не развивается.

С 2014 Microsoft стал развивать альтернативную платформу - .NET Core, которая уже предназначалась для разных платформ и должна была вобрать в себя все возможности устаревшего .NET Framework и добавить новую функциональность. Поэтому следует различать .NET Framework, который предназначен преимущественно для Windows, и кроссплатформенный .NET Core.

Также стоит упомянуть о платформе Mono, которая была создана еще в 2004 году и представляла опенсорс-версию платформы .NET Framework для Linux и MacOS. Используя Mono, можно было создавать кроссплатформенные приложения на C#.

# JIT-КОМПИЛЯЦИЯ

Код на C# компилируется в приложения или сборки с расширениями exe или dll на языке CIL (Common Intermediate Language) - своего рода ассемблер платформы .NET.

Далее при запуске на выполнение подобного приложения происходит JIT-компиляция (Just-In-Time) в машинный код, который затем выполняется. При этом, поскольку приложение может быть большим и содержать кучу инструкций, в текущий момент времени будет компилироваться лишь та часть приложения, к которой непосредственно идет обращение.

Если обратиться к другой части кода, то она будет скомпилирована из CIL в машинный код. При том уже скомпилированная часть приложения сохраняется до завершения работы программы. В итоге это повышает производительность.